Classification and prediction for the breast cancer tumors

Report prepared by Xinyi Xie, Jiawei Zeng and Kexin Fan

Professor: Jay Newby

MATH 509

December 8, 2022

Table of Contents

Background

Cancer is a disease in which cells in the body grow out of control. Breast cancer is one of the most common cancers diagnosed. Many of us think of breast cancer as a female disease, but it can also occur in men. We filtered out some of the features in the progress of searching data. Based on these features, we can determine whether a tumor is benign(B) or malignant(M).

Objective

Our objective of this project is to investigate the most optimal way to classify tumors as benign and malignant, provide an understanding of the process of organizing and preparing data, selecting features and applying machine learning tools. We will mainly focus on the random forest and adaboost methods to determine the accuracy of the tumor classification from various aspects, select and also give a general judgment that can be applied to the prediction method of tumors in the future.

Dataset preparation

1. Clean and Generate data

We choose diagnosis as attribute factors and 10 real-valued features to predict:

1. Radius (mean of distances from center to points on the perimeter)
2. Texture (standard deviation of gray-scale values)
3. Perimeter
4. Area
5. Smoothness (local variation in radius lengths)

6. Compactness (perimeter^2 / area - 1.0)

7. Concavity (severity of concave portions of the contour)

8. concave points (number of concave portions of the contour)

9. symmetry

10. fractal dimension(coastline approximation - 1)

After modifying the data frame, we got a data quality report, there is no missing value,

Response variable:

	Count	Miss %	Card.	Mode	Mode Freq	Mode %	2nd Mode	2nd Mode Freq	2nd Mode %
diagnosis	569	0.0	2	NaN	357	62.741652	NaN	212	37.258348

10 features:

	Count	Miss %	Card.	Min	1st Qrt.	\
Radius_mean	569	0.0	456	6.981000	11.7	
Texture_mean	569	0.0	479	9.710000	16.17	
perimeter_mean	569	0.0	522	43.790000	75.17	
area_mean	569	0.0	539	143.500000	420.3	
smoothness_mean	569	0.0	474	0.052630	0.08637	
compactness_mean	569	0.0	537	0.019380	0.06492	
concavity_mean	569	0.0	537	0.000000	0.02956	
concave points_mean	569	0.0	542	0.000000	0.02031	
symmetry_mean	569	0.0	432	0.106000	0.1619	
fractal_dimension_mean	569	0.0	499	0.049960	0.0577	

Irregular cardinality:

The following common issues such as the features with a cardinality of 1, too high or low cardinality for categorical features do not appear in the dataframe, so we do not need to drop any column.

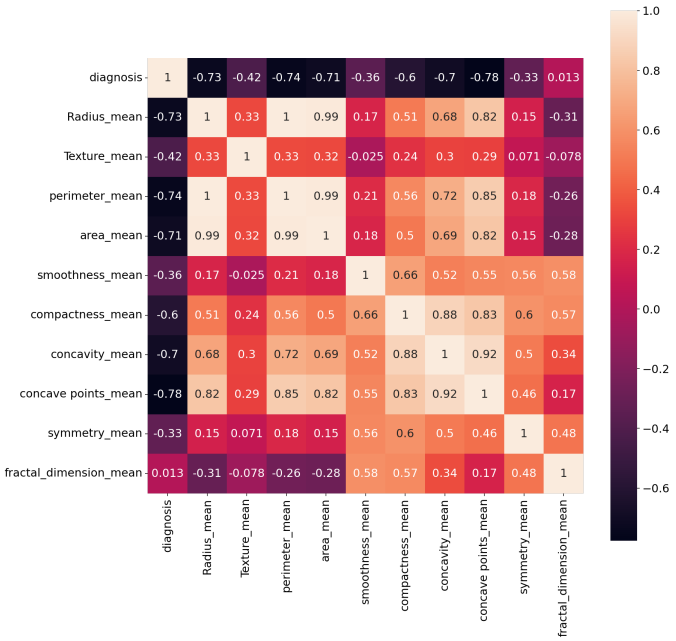
```
569
diagnosis                 2
Radius_mean             456
Texture_mean            479
perimeter_mean          522
area_mean               539
smoothness_mean         474
compactness_mean        537
concavity_mean          537
concave points_mean     542
symmetry_mean           432
fractal_dimension_mean  499
dtype: int64
```

We did an overall summary of data:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              diagnosis   R-squared (uncentered):                   0.255
Model:                            OLS   Adj. R-squared (uncentered):              0.241
Method:                 Least Squares   F-statistic:                              19.08
Date:                Mon, 05 Dec 2022   Prob (F-statistic):                    2.57e-30
Time:                        03:23:58   Log-Likelihood:                         -591.20
No. Observations:                 569   AIC:                                      1202.
Df Residuals:                     559   BIC:                                      1246.
Df Model:                          10
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
Radius_mean               -1.7160      1.160     -1.480      0.139      -3.994       0.562
Texture_mean              -0.0949      0.032     -3.011      0.003      -0.157      -0.033
perimeter_mean             1.3256      1.279      1.036      0.301      -1.187       3.839
area_mean                  0.3363      0.217      1.550      0.122      -0.090       0.763
smoothness_mean           -0.0276      0.050     -0.555      0.579      -0.125       0.070
compactness_mean          -0.0052      0.138     -0.038      0.970      -0.275       0.265
concavity_mean            -0.0646      0.099     -0.653      0.514      -0.259       0.130
concave points_mean       -0.2495      0.135     -1.851      0.065      -0.514       0.015
symmetry_mean             -0.0275      0.039     -0.713      0.476      -0.103       0.048
fractal_dimension_mean     0.0016      0.074      0.022      0.983      -0.143       0.146
==============================================================================
Omnibus:                       13.853   Durbin-Watson:                   0.258
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               14.511
Skew:                          -0.384   Prob(JB):                     0.000706
Kurtosis:                       2.846   Cond. No.                         139.
==============================================================================
```

From this summary, the p-value of texture mean is the smallest which is 0.003, far smaller than the p-value of other features. And concave points mean p-value 0.065 is another feature in our summary. Therefore, texture mean and concave points mean are significant features during our pre-processing phase.

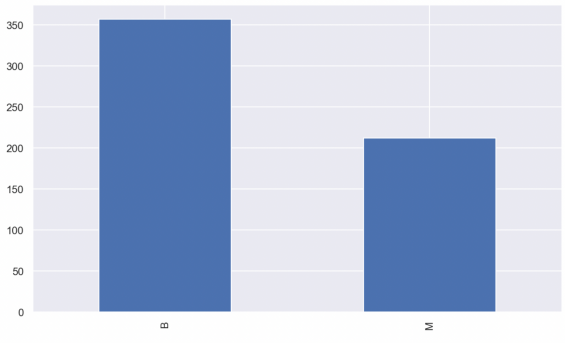
2. Visualizing data



We use Seaborn to create a heat map of the correlations between the features. From this heat map, we can get the relationship between diagnosis and other features, the correlation is higher when it is much closer to 1.0.

According to the output of the heatmap, there are several variables that are highly correlated. They are area and radius, perimeter and area, and compactness and concavity. The strong correlation between the first two groups of variables may be due to the inclusion of each other in their calculation formulas. There may be a collinear relationship between the latter set of variables.

3. Barplot



There are 212 malignant(M) and 357 benign(B). So, we do not need to do a subsample since the quantity of the target feature is almost balanced.

Method

We choose an ensemble machine learning method to do our project. Ensemble learning is a method of generating various base classifiers from which new classifiers are derived that perform better than any of the constituent classifiers. The word ensemble is a Latin-derived word which means ‘union of parts’. In this method, we first divide our whole dataset into two parts: training set and test set. Then we learn multiple alternative models using different training data or different learning algorithms. Finally we will do the voting and combine decisions of multiple models and get the final model. The key objective of the ensemble methods is to reduce bias and variance.

We set 70% of data as the training set and 30% as the testing set based on all data as our models. All our methods used belong to the ensemble algorithm. Ensemble learning refers to algorithms that combine the predictions from two or more models. It is a way of enlarging the hypothesis space.

Bagging is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It learns multiple hypotheses from different random subsets of the training set, then takes a majority vote.

$$e= (\Sigma\ e_i)/n$$

where e_1,e_2…..e = base classifier, e = final classifier

Boosting is another method used in machine learning to reduce errors in predictive data analysis. It learns by assigning a weight for various items in the data. The boosting technique initially starts with equal weights but after every model, each model is assigned a weight based on its performance. If the observation is classified incorrectly, the weight of observation will increase. Otherwise, the weight of observation will decrease.

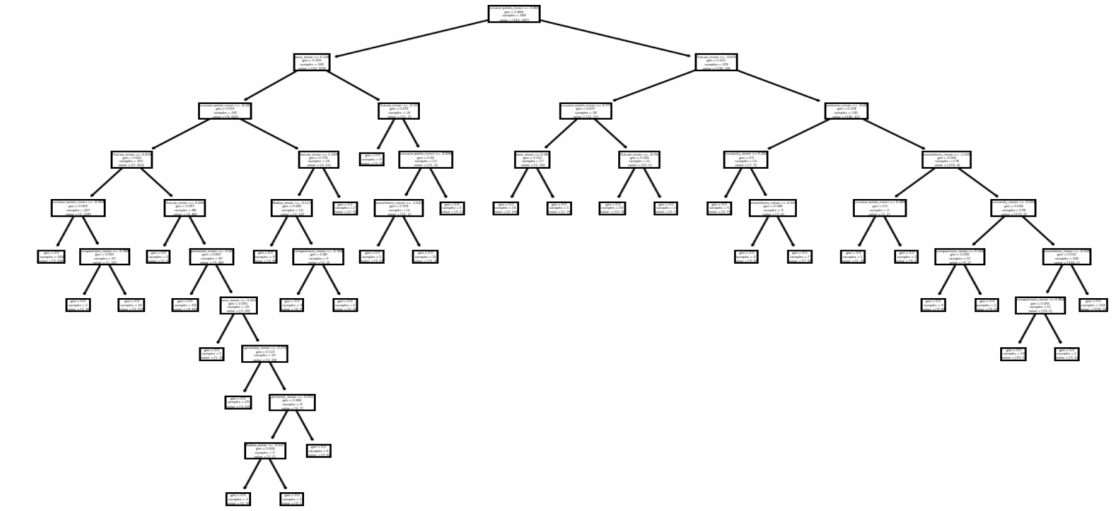
$$e= ((\Sigma\ e_i w_i)/*\Sigma\ w_i))/n$$

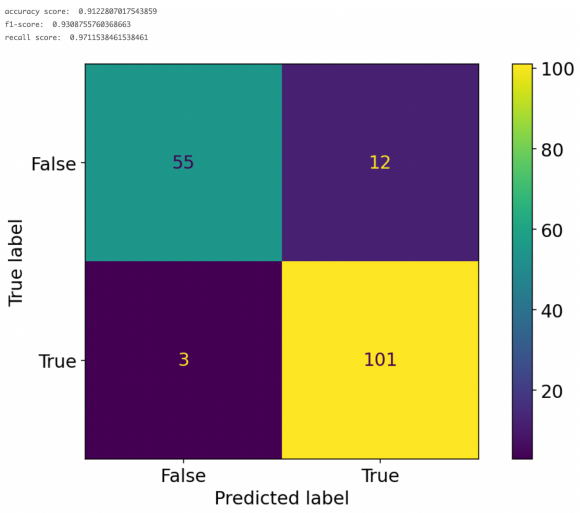
where, e_1,e_2…..e = base classifier, w_1,w_2…..w =weight, n=no. of models, e = final classifier

The difference between bagging and boosting classifiers are majority on training of base classifier and voting. First, each training in bagging is done independently and parallel, but in boosting, the next training set relies on previous training. The second one is voting. For the bagging, the final predictions are determined by combining the predictions from all the models. But for boosting, the final model is derived from various models that look at different sets of data, voting on them based on their weight. In the last part, we will use random forest as an example of a bagging method, and Adaboost as an example of the boosting method to classify data.

1. Decision tree

Decision trees perform a greedy search to identify the best split points within the decision tree. This splitting process is then repeated in a top-down regression fashion until all or most records are labeled with a particular class label.





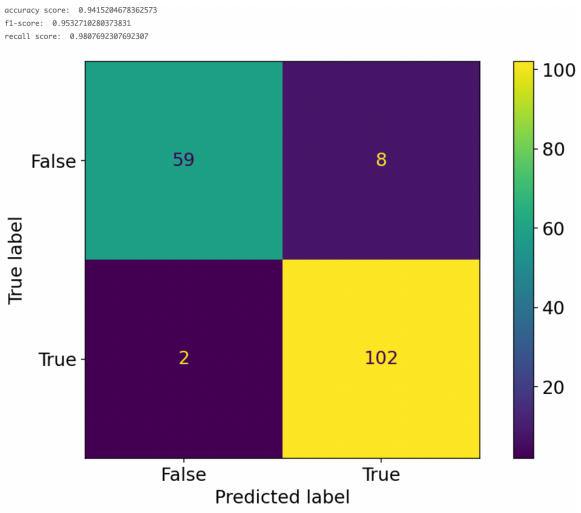
First, we made a confusion matrix of the results of the test data to allow us to observe the predicted results more intuitively. For our data, the yellow part proves that it is actually benign, and the predicted value is also benign; the green part is actually malignant, and the predicted

result is also malignant; the purple part in the upper right corner indicates that it is actually malignant, and the predicted value is benign; the lower left purple area of the corner indicates that the actual value is benign and the predicted value is malignant. Therefore, we can more intuitively see that the cases who are correctly predicted far outnumber those who are incorrectly predicted.

Then, we can calculate the mean accuracy of the predictions of test sets. The accuracy score is the number of correct results divided by the number of testing observations which is 91.228%, and recall score the number predicted to be benign divided by the number that was actually benign which is 0.97115, which is considerably good as it is above 0.5. So, the results of the decision tree are not bad, and we will use the random forest method to redo it.

2. Random forest

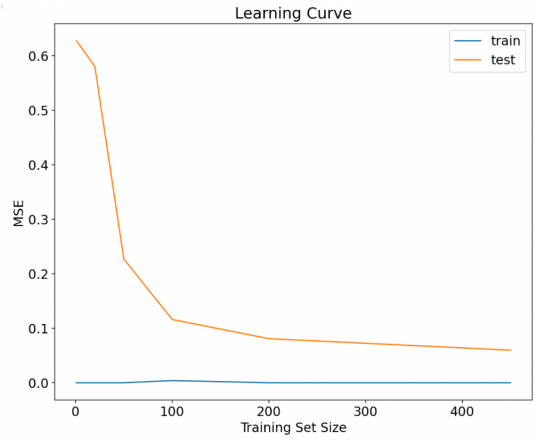
a). Confusion matrix



As a decision tree algorithm to be the base classifier, we set 30 trees to do the random forest. The accuracy score is 94.152%, and the recall score is 0.98077. Then we calculate the variance and

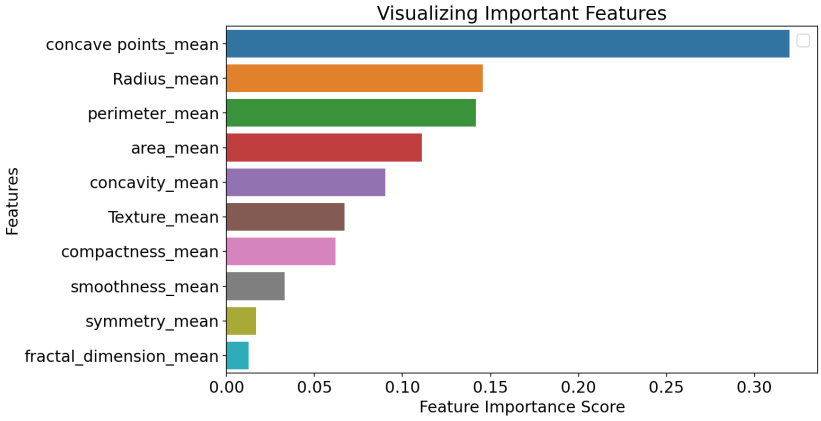
bias is 0.22947 and 0.4790327 respectively. Both of them are relatively low, which means that our model is not too bad. We also calculate the bias value is 0.479, and the variance is 0.2295. Compared with the decision tree, the random forest method is better than it.

b). Learning curve



From the learning curve, we can see that when training set size at 200, the value of MSE began to show a trend towards a steady decline around 0.08, which is small enough, so when our testing set is about 35%, we can get a relatively high accuracy value.

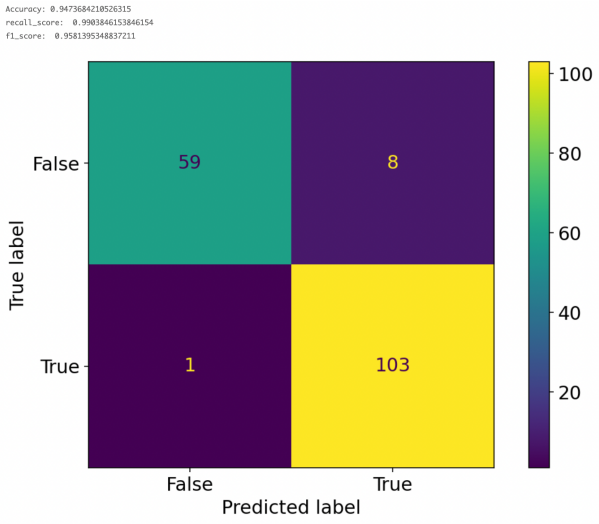
c). Feature selection



We did feature selection based on random forest. When we remove misleading data and noise, reduce 3 less important variables which are less than 0.05 (smoothness, fractal dimension and symmetry) , it improves our accuracy and also reduces the training time from 1.3s to 0.7s. The accuracy score is 0.9532, and recall score is 0.9455. This is more effective for our model than the original one.

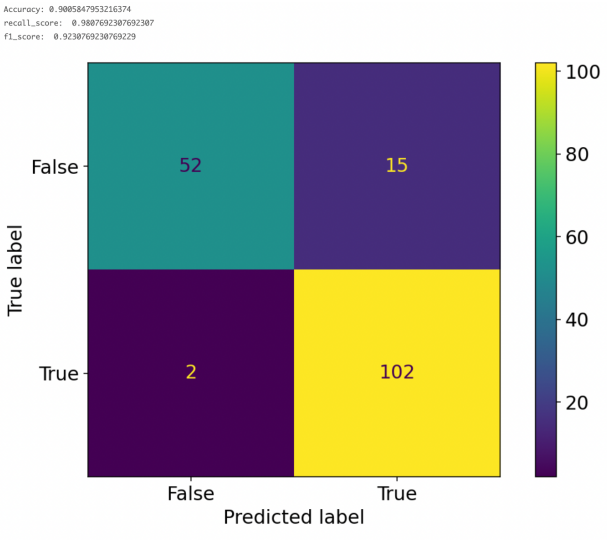
2. Ada boost

a). Confusion matrix



(Based decision tree)

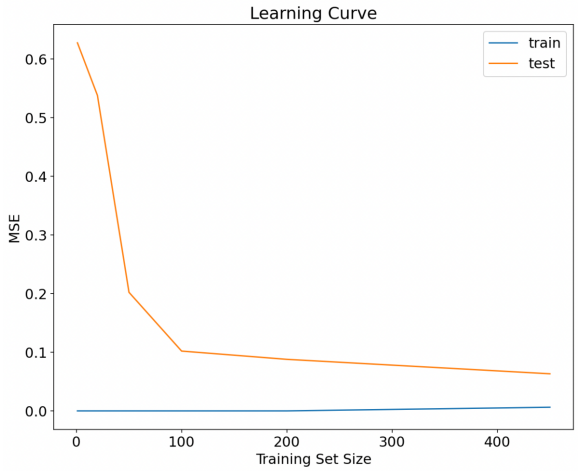
We use this method based on our original features. The accuracy score is 0.94737 which means we have 94.737% accuracy, higher than the accuracy score of random forest. And the recall score is 0.99 which is a very good value. The bias is 0.4772, and the variance is 0.2278. It shows that the ada boost is more effective than the random forest method before feature selection.



(Based support vector classifier(SVM))

Support vector machine algorithm is to find a hyperplane in an N-dimensional space (N: the number of features) that distinctly classifies the data points. It aims to do this compared with the decision tree. The accuracy score is 90.058% and recall score is 0.98. Comparing these two values with previous ones based on the decision tree, they are especially not as good as the previous one, thus the accuracy score is the lowest.

b). Learning curve



Not same as the result in the random forest method, when training set size at 100, the value of MSE began to show a trend towards a steady decline around 0.1, so when our testing set is about 20%, we can get a relatively high accuracy value.

Exploring setting

It is interesting that at first we randomly set the number of estimated classifier trees to be 50, we generally believe that the number of trees is better when we have enough large size. Then secondly we tried to set the number of estimated classifier trees to be 30. After comparing the accuracy score and recall score under these two situations, we found when the number of trees is 30 will get higher accuracy than 50.

We speculated whether the smaller the number of trees set, the higher the accuracy. However, when we set the tree number to be 20, we found that the result of 30 is still better than 20. Therefore, we conclude that when we set the number of trees randomly, what we should be setting is the appropriate number, not the larger the better.

Conclusion

After detailed classification, comparing the bias and variance, and the accuracy of prediction, we got the adaboost method with the decision tree is the best for us to classify whether the tumors are benign or malignant. Comparing all the methods we used, the boosting classifier got more recall score than the bagging classifier. It is more accurate to predict whether the tumor is benign.

Reference

Acharya, T. (2019, June 24). *Advanced ensemble classifiers*. Medium. Retrieved December 8, 2022, from https://towardsdatascience.com/advanced-ensemble-classifiers-8d7372e74e40

Bagging vs boosting in Machine Learning: Difference between bagging and boosting. upGrad blog. (2022, October 27). Retrieved December 8, 2022, from https://www.upgrad.com/blog/bagging-vs-boosting/#:~:text=Boosting%20decreases%20bias%2C%20not%20variance,are%20built%20independently%20in%20Bagging.

Learning, U. C. I. M. (2016, September 25). *Breast cancer wisconsin (diagnostic) data set*. Kaggle. Retrieved December 8, 2022, from https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data